Linear Programming Decoding of Binary Linear Codes for Symbol-Pair Read Channels

Shunsuke Horii, Toshiyasu Matsushima, and Shigeichi Hirasawa

Abstract

In this paper, we develop a new decoding algorithm of a binary linear codes for symbol-pair read channels. Symbol-pair read channel has recently been introduced by Cassuto and Blaum to model channels whose write resolution is higher than its read resolution. The proposed decoding algorithm is based on a linear programming (LP). It is proved that the proposed LP decoder has the maximum-likelihood (ML) certificate property, i.e., the output of the decoder is guaranteed to be the ML codeword when it is integral. We also introduce the *fractional pair distance* d_{fp} of a code which is a lower bound on the pair distance. It is proved that the proposed LP decoder will correct up to $\lceil d_{fp}/2 \rceil - 1$ pair errors.

I. INTRODUCTION

Recent advances in data storage technologies make it possible to write information to storage devices with high resolution. However, it causes a problem of the gap between write resolution and read resolution. In the case that read resolution is lower than write resolution, it is difficult to successfully read the individual symbols written to the storage devices. The *symbol pair read channel* proposed by Cassuto and Blaum [1] is the channel model that models reading process of such storage devices. On each reading operation, two consecutive symbols are read. We need new error-correction framework for the new channel model. In the reading process, symbol-pair errors would occur rather than individual symbol errors, where a symbol-pair error means at least one of the symbol pair is read erroneously. The main task is to construct error correcting codes and decoding algorithms that can correct many symbol-pair errors as possible.

Some error correcting codes and decoding algorithms have been studied. Cassuto and Blaum [1] built a framework for correcting pair errors, studying a new metric called pair-distance and its relation to pair error correction capability. They also provided code constructions and decoding algorithms and derived some bounds on the size of optimal symbol-pair codes. Cassuto and Listsyn [2] proposed algebraic construction of cyclic symbol-pair codes and studied its property. Yaakobi et al. [3] proposed efficient decoding algorithms for the cyclic symbol-pair codes. Chee et al. [4] established a Singleton-type bound for symbol-pair codes and constructed codes that meet the Singleton-type bound. Hirotomo et al. [5] proposed the decoding algorithm for symbol-pair codes based on the newly defined parity-check matrix and syndromes.

In this paper, we develop a new decoding algorithm of a binary linear codes for symbol-pair read channels. We establish a decoding algorithm which is based on a linear programming (LP). The LP based decoding algorithm

(LP decoder) of a binary linear codes for memoryless channels was originally proposed by Feldman et al. [6] and studied further by many researchers. The LP decoder has many attractive properties as follows.

- The output of the decoder is guaranteed to be the maximum-likelihood (ML) codeword (ML certificate property) [6].
- LP decoding with expander codes achieves the capacity of any binary-input memoryless symmetric LLR-bounded (MSB) channel [7].
- There exist algorithms which efficiently solve the LP [8][9].

It is natural to expect that the decoder based on the LP also works well for symbol-pair read channel. Not only establishing the LP decoder for symbol-pair read channel, we also investigate its property. We show that the LP decoder has ML certificate property even when the channel is symbol-pair read channel. We also introduce the *fractional pair distance* d_{fp} of a code which is a lower bound on the pair distance. It is proved that the proposed LP decoder will correct up to $\lceil d_{fp}/2 \rceil$ errors. We examine the average fractional pair distance of randomly chosen LDPC codes from an ensemble of Gallager [10].

The rest of the paper is organized as follows. In Section 2, we introduce basic notation and definitions for symbol-pair read channels. In Section 3, we establish the LP decoder for symbol-pair read channels and in Section 4, some properties of the proposed LP decoder are given. We conclude the paper in Section 5.

II. PRELIMINARIES

In this section, we briefly review the model and definition of the symbol-pair read channel and error-correcting codes. Throughout this paper, we deal with binary linear codes and let codeword symbols $\Sigma = \{0, 1\}$. Let n denotes the code length and $\mathcal{C} \subset \Sigma^n$ denotes a binary linear code. For a length n vector $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \Sigma^n$, the symbol-pair read vector of $\boldsymbol{x}$, denoted by $\pi(\boldsymbol{x})$, is defined as

$$\pi(\boldsymbol{x}) = ((x_0, x_1), (x_1, x_2), \cdots, (x_{n-2}, x_{n-1}), (x_{n-1}, x_0)).$$

Unless stated otherwise, all indices are taken modulo n. The pair-distance between vectors $\boldsymbol{x} \in \Sigma^n$ and $\boldsymbol{y} \in \Sigma^n$ is defined as

$$D_p(\boldsymbol{x}, \boldsymbol{y}) = D_H(\pi(\boldsymbol{x}), \pi(\boldsymbol{y}))$$
$$= |\{i : (x_i, x_{i+1}) \neq (y_i, y_{i+1})\}|,$$

where D_H denotes the usual Hamming distance. Similarly, the pair weight of $\boldsymbol{x}$ is defined as

$$W_p(\boldsymbol{x}) = W_H(\pi(\boldsymbol{x}))$$
$$= |\{i : (x_i, x_{i+1}) \neq (0, 0)\}|,$$

where W_H denotes the usual Hamming weight. We say that symbol-pair error is occurred at i-th position when at least one of two bits (x_i, x_{i+1}) are changed during reading operation.

We assume the probabilistic model for symbol-pair read channel. Let $p < \frac{3}{4}$ be the probability that the symbol-pair error occurs and let

$$p(\boldsymbol{b}|\boldsymbol{a}) = \begin{cases} 1-p & \text{if } \boldsymbol{a} = \boldsymbol{b} \\ \frac{p}{3} & \text{otherwise} \end{cases} \quad \boldsymbol{a},\boldsymbol{b} \in \{0,1\}^2 .$$

Let $\overset{\leftrightarrow}{\boldsymbol{y}} \in (\Sigma \times \Sigma)^n$ be the received pair vector. We assume that pair errors occur independently at each position i and

$$p(\overset{\leftrightarrow}{\boldsymbol{y}}|\boldsymbol{x}) = p(\overset{\leftrightarrow}{\boldsymbol{y}}|\pi(\boldsymbol{x}))$$

$$= \prod_{i=0}^{n-1} p(\overset{\leftrightarrow}{\boldsymbol{y}}_i|(\pi(\boldsymbol{x}))_i), \tag{1}$$

where $\overset{\leftrightarrow}{\boldsymbol{y}}_i$ and $(\pi(\boldsymbol{x}))_i$ are the ith symbol-pair of $\overset{\leftrightarrow}{\boldsymbol{y}}$ and $\pi(\boldsymbol{x})$, respectively. We call $D_H(\pi(\boldsymbol{x}), \overset{\leftrightarrow}{\boldsymbol{y}})$ the number of pair errors when the $\boldsymbol{x}$ is the sent codeword and $\overset{\leftrightarrow}{\boldsymbol{y}}$ is the received pair vector.

III. LP DECODING FOR SYMBOL-PAIR READ CHANNEL

When $\overset{\leftrightarrow}{\boldsymbol{y}}$ is the read pair vector, ML decoding problem is described as

$$\begin{aligned} \text{minimize} \quad & -\ln p(\overset{\leftrightarrow}{\boldsymbol{y}}|\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathcal{C}. \end{aligned} \tag{2}$$

This problem includes the ML decoding problem for memoryless channels and therefore it is NP-hard in general [11]. Here, we derive the relaxed problem.

Let $\tau_{i,(a,b)}(\boldsymbol{x})$ be the indicator function which represents whether $(\pi(\boldsymbol{x}))_i = (a,b)$ or not, i.e.,

$$\tau_{i,(a,b)}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } (\pi(\boldsymbol{x}))_i = (a,b) \\ 0 & \text{otherwise} \end{cases} .$$

We also define the map $T : \Sigma^n \to \{\Sigma^2\}^n$ as $T(\boldsymbol{x}) = \left(\left(\tau_{i,(a,b)}(\boldsymbol{x}) \right)_{(a,b)\in\{\Sigma^2\}} \right)_{i=0,\dots,n-1}$. Note that $T : \Sigma^n \to \{\Sigma^2\}^n$ is an injection and the relations $x_i = \tau_{i,(1,0)}(\boldsymbol{x}) + \tau_{i,(1,1)}(\boldsymbol{x})$, $x_{i+1} = \tau_{i,(0,1)}(\boldsymbol{x}) + \tau_{i,(1,1)}(\boldsymbol{x})$ hold.

From the assumption for the channel, $-\ln p(\overset{\leftrightarrow}{\boldsymbol{y}}|\boldsymbol{x})$ can be written as $\sum_{i=0}^{n-1} -\ln p(\overset{\leftrightarrow}{\boldsymbol{y}}_i|\pi(\boldsymbol{x})_i)$. Given $\overset{\leftrightarrow}{\boldsymbol{y}}_i$, $p(\overset{\leftrightarrow}{\boldsymbol{y}}_i|(\pi(\boldsymbol{x}))_i)$ is a function of $(\pi(\boldsymbol{x}))_i$. Let $\lambda_{i,(a,b)} = -\ln p(\overset{\leftrightarrow}{\boldsymbol{y}}_i|(a,b))$, $(a,b) \in \Sigma^2$ then,

$$-\ln p(\overset{\leftrightarrow}{\boldsymbol{y}}_i|(\pi(\boldsymbol{x}))_i) = \sum_{(a,b)\in\Sigma^2} \lambda_{i,(a,b)} \tau_{i,(a,b)}(\boldsymbol{x}).$$

We define $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_0, \cdots, \boldsymbol{\lambda}_{n-1})$ $\boldsymbol{\lambda}_i = (\lambda_{i,(a,b)})_{(a,b)\in\Sigma^2}$, $i = 0, \cdots, n-1$, and let Λ be the map from $\boldsymbol{y}$ to $\boldsymbol{\lambda}$ i.e., $\boldsymbol{\lambda} = \Lambda(\boldsymbol{y})$.

Summarizing above, the optimization problem (2) can be converted to the following optimization problem.

$$\begin{aligned} \text{minimize} \quad & \langle \boldsymbol{\lambda}, \boldsymbol{\tau}(\boldsymbol{x}) \rangle \\ \text{s.t.} \quad & \tau_{i,(a,b)}(\boldsymbol{x}) \in \{0,1\}, \quad i = 0,\dots,n-1, (a,b) \in \Sigma^2 \\ & \sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)}(\boldsymbol{x}) = 1, \quad i = 0,\dots,n-1 \\ & x_i = \tau_{i,(1,0)}(\boldsymbol{x}) + \tau_{i,(1,1)}(\boldsymbol{x}), \quad i = 0,\dots,n-1 \\ & x_{i+1} = \tau_{i,(0,1)}(\boldsymbol{x}) + \tau_{i,(1,1)}(\boldsymbol{x}), \quad i = 0,\dots,n-1 \\ & \boldsymbol{x} \in \mathcal{C}, \end{aligned} \tag{3}$$

where $\tau(x) = T(x)$ and $\langle \lambda, \tau(x) \rangle = \sum_{i=0}^{n-1} \sum_{(a,b) \in \Sigma^2} \lambda_{i,(a,b)} \tau_{i,(a,b)}(x)$. Note that the constraints other than $x \in \mathcal{C}$ are redundant constraints that are automatically satisfied if $x \in \mathcal{C}$.

We only rewrote the problem (2) in another form and therefore it is still hard to solve the optimization problem. Then, we relax the constraints $\tau_{i,(a,b)} \in \{0,1\}$ to $\tau_{i,(a,b)} \in [0,1]$ and $x \in \mathcal{C}$ to $x \in \mathcal{Q}(H)$, where H is the parity check matrix of $\mathcal{C}$ and $\mathcal{Q}(H)$ is the fundamental polytope of $\mathcal{C}$ [6]. $\mathcal{Q}(H)$ has the following properties [6]

- $\mathcal{C} \subset \mathcal{Q}(H)$
- $\mathcal{Q}(H) \cap \{0,1\}^n = \mathcal{C}$
- $x \in \mathcal{Q}(H)$ can be expressed by $2n + \sum_{j=1}^{m} 2^{d_j - 1}$ inequalities (m is the number of rows of H and d_j is the Hamming weight of the ith row of H)

The relaxed problem is described as follows.

$$
\begin{aligned}
\text{minimize} \quad & \langle \lambda, \tau \rangle \\
\text{s.t.} \quad & \tau_{i,(a,b)} \in [0,1], \quad i = 0, \ldots, n-1, (a,b) \in \Sigma^2 \\
& \sum_{(a,b) \in \Sigma^2} \tau_{i,(a,b)} = 1, \quad i = 0, \ldots, n-1 \\
& x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}, \quad i = 0, \ldots, n-1 \\
& x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}, \quad i = 0, \ldots, n-1 \\
& x \in \mathcal{Q}(H).
\end{aligned}
\tag{4}
$$

Note that $\tau = (\tau_0, \cdots, \tau_{n-1})$, $\tau_i = (\tau_{i,(a,b)})_{(a,b) \in \Sigma^2}$ are the optimization variables that are independently defined from x. If we use LDPC code for $\mathcal{C}$, (4) is solvable in the polynomial order of n.

The LP decoder for symbol-pair read channel consists of the following steps.

1. Calculate λ from $\overset{\leftrightarrow}{y}$.

2. Solve the LP (4).

3. If the optimal solution x^* is integral ($x^* \in \{0,1\}^n$), output x^* and declare "error" otherwise.

As in the case for the memoryless channel, the following theorem holds.

Theorem 1: If the LP decoder outputs a codeword, it is guaranteed to be the ML codeword.

Proof: Let (x^*, τ^*) be the optimal solution of (4). Its objective value is smaller than or equal to that of other feasible solution (x, τ). For any codeword $x' \in \mathcal{C}$, it is easily verified that $(x', T(x'))$ is a feasible solution for (4). In the following, we will confirm that no other feasible solution of the form (x', τ) exists.

First, if x' is integral, τ must be integral because assumption that τ is fractional leads to the contradiction. For example, if $0 < \tau_{i,(1,0)} < 1$ then $0 < \tau_{i,(1,1)} < 1$ must holds from the constraint $\tau_{i,(1,0)} + \tau_{i,(1,1)} = x_i$ and assumption that x_i is integral. Then $\tau_{i,(0,1)} = 1 - \tau_{i,(1,1)}$ from the constraint $\tau_{i,(0,1)} + \tau_{i,(1,1)} = x_{i+1}$ and assumption that x_{i+1} is integral. Then $\sum_{(a,b) \in \Sigma^2} \tau_{i,(a,b)} \geq \tau_{i,(0,1)} + \tau_{i,(1,0)} + \tau_{i,(1,1)} = 1 - \tau_{i,(1,1)} + \tau_{i,(1,0)} + \tau_{i,(1,1)} > 1$ and it contradicts the constraint $\sum_{(a,b) \in \Sigma^2} \tau_{i,(a,b)} = 1$. We can lead contradiction in other cases similarly.

Further, there doesn't exist integral τ such that (x', τ) is a feasible solution of (4) unless $\tau = T(x')$ because $T : \Sigma^n \to \{\Sigma^2\}^n$ is an injection.

Therefore (x^*, τ^*) has cost less than ore equal to feasible point that corresponds to other codeword x'. ∎

We will show later in Lemma 1 that if (x^*, τ^*) is a vertex of the feasible region of (4) then x^* is the vertex of $\mathcal{Q}(H)$. When the factor graph of the code is cycle-free, $\mathcal{Q}(H)$ has no fractional vertex [6]. Therefore the

proposed optimization problem (4) is an exact formulation of the ML decoding problem of the binary linear code for symbol-pair read channels in the cycle-free case.

IV. ANALYSIS OF LP DECODING FOR SYMBOL-PAIR READ CHANNEL

Let $\Pr\left[\text{err}|\boldsymbol{x}\right]$ be the probability that the LP decoder makes an error, given that $\boldsymbol{x}$ was sent. It can be described as

$$\Pr\left[\text{err}|\boldsymbol{x}\right] = \Pr\left\{\exists(\boldsymbol{x}', \boldsymbol{\tau}') \text{ s.t } (\boldsymbol{x}', \boldsymbol{\tau}') \text{ is a feasible solution of (4)}, \langle \boldsymbol{\lambda}, \boldsymbol{\tau}' \rangle \leq \langle \boldsymbol{\lambda}, T(\boldsymbol{x}) \rangle \right\}. \tag{5}$$

Here, we analyze the performance of LP decoder for symbol-pair read channels.

A. The All-Zeros Assumption

It is common to assume that the transmitted codeword is the all-zeros vector, because the assumption makes it easy to analyze the performance of the decoder. Although it is not straight forward to verify that we can assume the assumption, we will prove that it is valid for the LP decoder for symbol-pair read channel as it is valid for memoryless channel [6].

Theorem 2: The error probability of the LP decoder for the symbol-pair read channel (1) is independent of the transmitted codeword.

Proof: See Appendix A. ■

According to the theorem, we can assume that the transmitted codeword is all-zeros vector $\boldsymbol{0}$.

B. Fractional Pair Distance

Here, we introduce the concept of *fractional pair distance* d_{fp} of a code $\mathcal{C}$ whose parity check matrix is H.

Definition 1: The fractional pair distance d_{fp} of a code $\mathcal{C}$ whose parity check matrix is H is defined as

$$d_{fp} = \min_{\boldsymbol{x} \in \mathcal{V}_{\mathcal{Q}} \backslash \{\boldsymbol{0}\}} W_{fp}(\boldsymbol{x}) \tag{6}$$

$$= \min_{\boldsymbol{x} \in \mathcal{V}_{\mathcal{Q}} \backslash \{\boldsymbol{0}\}} \sum_{i=0}^{n-1} w_{fp}((\pi(\boldsymbol{x}))_i), \tag{7}$$

where $\mathcal{V}_{\mathcal{Q}}$ is the set of vertices of the fundamental polytope $\mathcal{Q}(H)$ and

$$w_{fp}((a, b)) = \max\{a, b\} \tag{8}$$

The fractional distance of the code is defined as $d_f = \min_{\boldsymbol{x} \in \mathcal{V}_{\mathcal{Q}} \backslash \{\boldsymbol{0}\}} x_i$ in [6] and the fractional pair distance is an modified notion of it so that it is suitable for the symbol-pair read channel. As a trivial relation, $d_f \leq d_{fp}$ holds from the inequality $x_i \leq \max(x_i, x_{i+1}) = w_{fp}((\pi(\boldsymbol{x}))_i)$. The pair distance of a code $\mathcal{C}$ is defined as $d_p = \min_{\boldsymbol{x} \in \mathcal{C} \backslash \boldsymbol{0}} W_p(\boldsymbol{x})$ [1]. It holds that $d_{fp} \leq d_p$ from the fact that $\mathcal{C} \subset \mathcal{V}_{\mathcal{Q}}$ and $w_{fp}((a, b)) \leq 1$.

The performance of the LP decoder is characterized by the fractional pair distance as stated in the following theorem.

Theorem 3: For a code $\mathcal{C}$ with fractional pair distance d_{fp}^*, the LP decoder can correct up to $\lceil d_{fp}^*/2 \rceil - 1$ pair errors occurred through the channel (1).

Before proving the theorem, we prove the following lemma.

Lemma 1: Let $(\boldsymbol{x}, \boldsymbol{\tau})$ be a vertex of the feasible region of (4). Then $\boldsymbol{x}$ is a vertex of the fundamental polytope $\mathcal{Q}(H)$, i.e., $\boldsymbol{x} \in \mathcal{V}_{\mathcal{Q}}$.

Proof: See Appendix B. ∎

We are now ready to prove theorem 3.

Proof: (*Theorem 3*) We can assume that the transmitted codeword is $\boldsymbol{0}$ from theorem 2. The decoding failure of LP decoder means that the optimal solution $(\boldsymbol{x}^*, \boldsymbol{\tau}^*)$ satisfies $\boldsymbol{x}^* \neq \boldsymbol{0}$. From the assumption that the code has the fractional pair distance d^*_{fp} and Lemma 1, it holds that $\sum_{i=0}^{n-1} w_{fp}((\pi(\boldsymbol{x}^*))_i) \geq d^*_{fp}$. Let $\mathcal{E} = \left\{ i \mid \overset{\leftrightarrow}{y}_i \neq (0,0) \right\}$ be the set of positions that pair errors occurred.

To $(\boldsymbol{0}, \boldsymbol{\tau}^0)$ be a feasible solution of (4), $\boldsymbol{\tau}^0 = T(\boldsymbol{0})$ must hold and $\tau^0_{i,(0,0)} = 1$, $\tau^0_{i,(0,1)} = \tau^0_{i,(1,0)} = \tau^0_{i,(1,1)} = 0$ for $i = 0, \ldots, n-1$. Since $(\boldsymbol{x}^*, \boldsymbol{\tau}^*)$ is the optimal solution of (4), it holds that

$$\langle \boldsymbol{\lambda}, \boldsymbol{\tau}^* \rangle \leq \langle \boldsymbol{\lambda}, \boldsymbol{\tau}^0 \rangle. \tag{9}$$

Subtracting the right-hand side from the left-hand side in (9), we obtain

$$\langle \boldsymbol{\lambda}, \boldsymbol{\tau}^* \rangle - \langle \boldsymbol{\lambda}, \boldsymbol{\tau}^0 \rangle = \sum_{i=0}^{n-1} \sum_{(a,b) \in \Sigma^2} \lambda_{i,(a,b)} (\tau^*_{i,(a,b)} - \tau^0_{i,(a,b)})$$

$$= \sum_{i \in \mathcal{E}} \sum_{(a,b) \in \Sigma^2} \lambda_{i,(a,b)} (\tau^*_{i,(a,b)} - \tau^0_{i,(a,b)}) + \sum_{i \notin \mathcal{E}} \sum_{(a,b) \in \Sigma^2} \lambda_{i,(a,b)} (\tau^*_{i,(a,b)} - \tau^0_{i,(a,b)}). \tag{10}$$

For $i \in \mathcal{E}$, it holds that

$$\lambda_{i,(a,b)} = \begin{cases} -\ln \frac{p}{3} & \text{if } (a,b) = (0,0) \\ -\ln(1-p) & \text{if } (a,b) = \overset{\leftrightarrow}{y}_i \\ -\ln \frac{p}{3} & \text{otherwise.} \end{cases}$$

Therefore the first term in (10) can be expanded as

$$\sum_{i \in \mathcal{E}} \left(-\ln \frac{p}{3}(\tau^*_{i,(0,0)} - \tau^0_{i,(0,0)}) - \ln(1-p)(\tau^*_{i,\overset{\leftrightarrow}{y}_i} - \tau^0_{i,\overset{\leftrightarrow}{y}_i}) - \sum_{(a,b) \in \Sigma^2 \setminus ((0,0) \cup \overset{\leftrightarrow}{y}_i)} \ln \frac{p}{3}(\tau^*_{i,(a,b)} - \tau^0_{i,(a,b)}) \right)$$

$$= \sum_{i \in \mathcal{E}} \left(-\ln \frac{p}{3}(\tau^*_{i,(0,0)} - 1) - \ln(1-p)\tau^*_{i,\overset{\leftrightarrow}{y}_i} - \sum_{(a,b) \in \Sigma^2 \setminus ((0,0) \cup \overset{\leftrightarrow}{y}_i)} \ln \frac{p}{3}\tau^*_{i,(a,b)} \right)$$

$$= \sum_{i \in \mathcal{E}} \left(- \sum_{(a,b) \in \Sigma^2 \setminus \overset{\leftrightarrow}{y}_i} \ln \frac{p}{3}\tau^*_{i,(a,b)} - \ln(1-p)\tau^*_{i,\overset{\leftrightarrow}{y}_i} \right) + |\mathcal{E}| \ln \frac{p}{3}$$

$$= \sum_{i \in \mathcal{E}} \left(-\ln \frac{p}{3}(1 - \tau^*_{i,\overset{\leftrightarrow}{y}_i}) - \ln(1-p)\tau^*_{i,\overset{\leftrightarrow}{y}_i} \right) + |\mathcal{E}| \ln \frac{p}{3}$$

$$= \sum_{i \in \mathcal{E}} \left(\ln \frac{p}{3(1-p)}\tau^*_{i,\overset{\leftrightarrow}{y}_i} \right). \tag{11}$$

For $i \notin \mathcal{E}$, it hods that

$$\lambda_{i,(a,b)} = \begin{cases} -\ln(1-p) & \text{if } (a,b) = (0,0) \\ -\ln \frac{p}{3} & \text{otherwise.} \end{cases} \tag{12}$$

Therefore the second term in (10) can be expanded as

$$\sum_{i \notin \mathcal{E}} \left(-\ln(1-p)(\tau_{i,(0,0)}^* - \tau_{i,(0,0)}^0) - \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \ln \frac{p}{3}(\tau_{i,(a,b)}^* - \tau_{i,(a,b)}^0) \right)$$

$$= \sum_{i \notin \mathcal{E}} \left(-\ln(1-p)(\tau_{i,(0,0)}^* - 1) - \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \ln \frac{p}{3}\tau_{i,(a,b)}^* \right)$$

$$= \sum_{i \notin \mathcal{E}} \left(\ln(1-p) \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^* - \ln \frac{p}{3} \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^* \right)$$

$$= \sum_{i \notin \mathcal{E}} \ln \frac{3(1-p)}{p} \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^*. \tag{13}$$

Multiplying (10) by $\left(\ln \frac{p}{3(1-p)} \right)^{-1}$, [1]

$$\sum_{i \in \mathcal{E}} \tau_{i:\overset{\leftrightarrow}{\mathcal{Y}}_i}^* - \sum_{i \notin \mathcal{E}} \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^* \le |\mathcal{E}| - \sum_{i \notin \mathcal{E}} \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^*$$

$$\le |\mathcal{E}| - \sum_{i=0}^{n-1} \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^* + \sum_{i \in \mathcal{E}} \sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^*$$

$$\le 2|\mathcal{E}| - d_{fp}^*$$

$$< 0, \tag{14}$$

where the first line follows form $\tau_{i:\overset{\leftrightarrow}{\mathcal{Y}}_i}^* \le 1$ and the third line follows from $\sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^* \le 1$ and

$$\sum_{(a,b) \in \Sigma^2 \setminus (0,0)} \tau_{i,(a,b)}^* \ge \max \left\{ \tau_{i,(1,0)}^* + \tau_{i,(1,1)}^*, \tau_{i,(0,1)}^* + \tau_{i,(1,1)}^* \right\}$$

$$= \max \left\{ x_i^*, x_{i+1}^* \right\}$$

$$= w_{fp}((\pi(\boldsymbol{x}^*))_i)$$

$$\ge d_{fp}^*. \tag{15}$$

The inequality (14) suggests that $(10) > 0$ and it contradicts (9). Therefore $\boldsymbol{x}^* = \boldsymbol{0}$. ∎

C. Computing the Lower Bound of the Fractional Pair Distance

In [6], the efficient way to compute the fractional distance of an LDPC code is presented. We can compute the lower bound of the fractional pair distance of an LDPC code in a similar way. The reason that only lower bound can be computed is explained later. To compute the fractional pair distance, we have to compute $\min_{\boldsymbol{x} \in \mathcal{V}_Q \setminus \{\boldsymbol{0}\}} W_{fp}(\boldsymbol{x})$. Note that $W_{fp}(\boldsymbol{x})$ is a convex function of $\boldsymbol{x}$ since it can be expressed as the maximum of linear functions. Let $\mathcal{F}$ be the set of facets in $\mathcal{Q}(H)$ which $\boldsymbol{0}$ doesn't sit on. We choose a facet in $\mathcal{F}$ and minimize $W_{fp}(\boldsymbol{x})$ over that facet. The obtained minimum is a lower bound of the values of $W_{fp}(\boldsymbol{x})$ of the vertices in the facet. The minimum value obtained over all facets in $\mathcal{F}$ is the lower bound of $W_{fp}(\boldsymbol{x})$ over all vertices other than $\boldsymbol{0}$.

---

[1]note that $\left(\ln \frac{p}{3(1-p)} \right)^{-1} < 0$ when $p < 0.75$
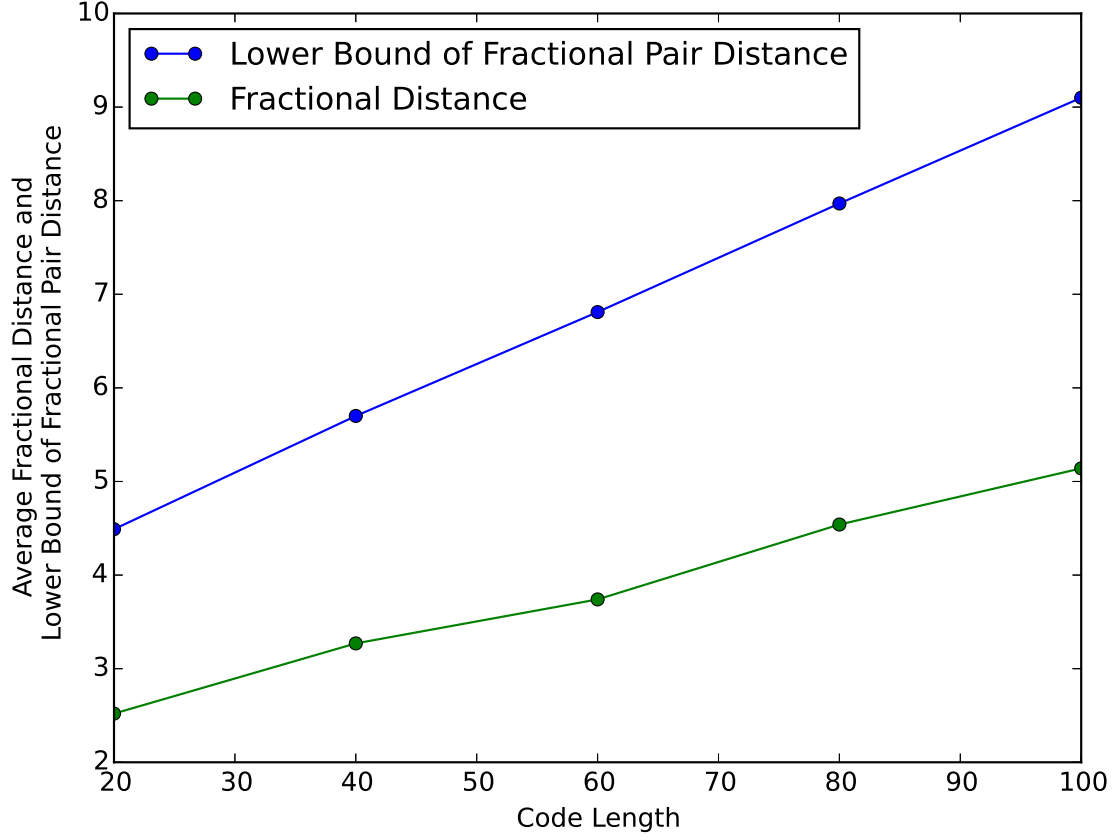
Fig. 1. The average fractional distance d_f and the lower bound of the fractional pair distance d_{fp} as functions of length for a randomly generated LDPC code, with variable degree 3, check degree 4, from an ensemble of Gallager [10].

Fig. 1 shows the average fractional pair distance of a randomly generated LDPC code. The generated LDPC code has variable degree 3, check degree 4, and is randomly generated from an ensemble of Gallager [10]. It seems that the lower bound of the fractional pair distance grows linearly with the block length. The fractional pair distance is about 1.8 times as large as the fractional distance. Since the fractional pair distance is a lower bound of the pair distance, the figure also gives a lower bound of the pair distance of the LDPC code.

D. Numerical Experiments

Here, we see the performance of the proposed LP decoder through numerical experiments. We used a randomly generated rate-1/4 LDPC code with variable degree 3 and check degree 4. As one benefit of formulating the decoding problem as an LP problem, we can obtain ML decoder by solving the corresponding integer linear program. Fig. 2 shows an error-rate comparison for a block length of 60. There exists a gap between the performance of LP decoder and that of ML decoder. We hope that this gap can be filled by tightening the relaxation as in the case for memoryless channels [6].
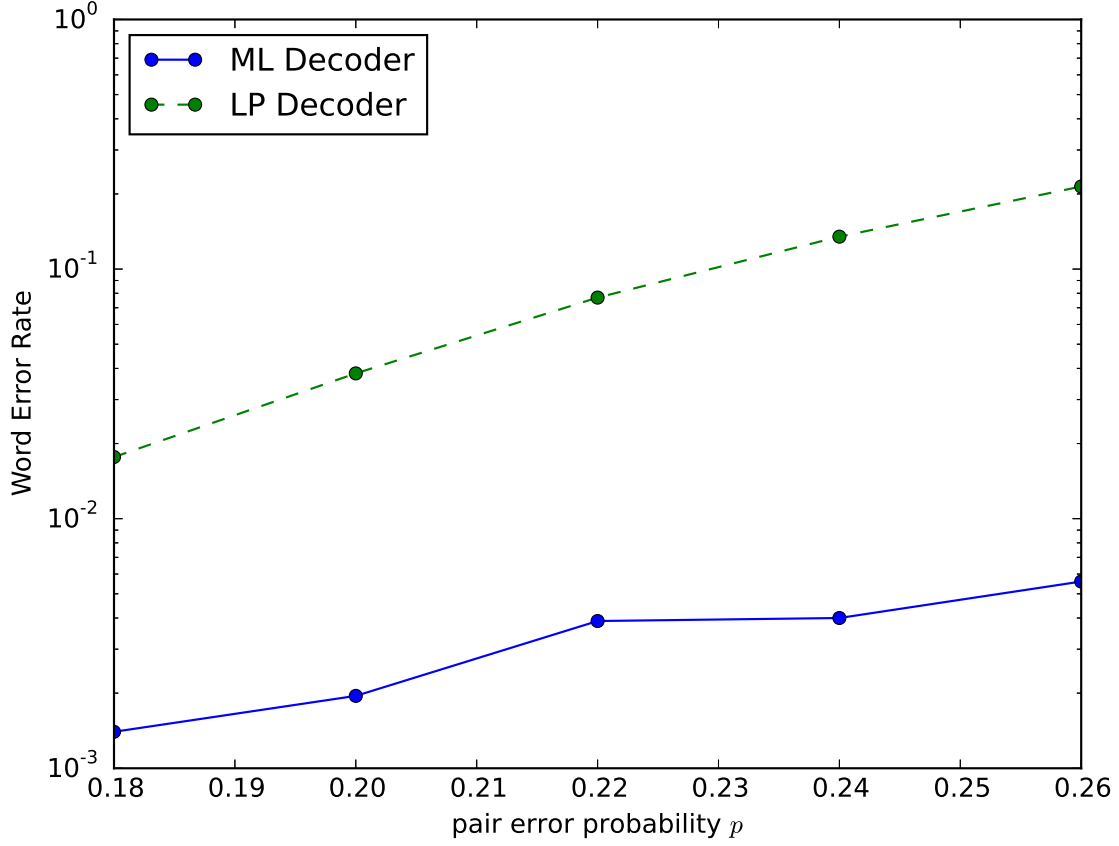
Fig. 2. A comparison between the performance of LP decoding and ML decoding using the same random rate-1/4 LDPC code with length 60.

V. CONCLUSION

We have developed an LP based decoding algorithm of binary linear codes for symbol-pair read channel. We also proved some results on its error-correcting performance. We introduced a notion called the fractional pair distance of a code, which characterize the lower bound of the number of correctable pair errors that the proposed algorithm can correct.

There are some future directions of the work presented here. We assumed that the pair error occurs at each position independently. However, this assumption may not hold for some cases. The LP decoding for finite-state channels is proposed in [13] and we can consider the symbol-pair read channel with finite state. To develop the LP decoding algorithm for such channel is a future work.

In this paper we assumed that the code is binary. The LP decoding for nonbinary linear codes for memoryless channel was presented in [14]. It would be interesting to develop the LP decoding algorithm of nonbinary linear codes for symbol-pair read channel.

APPENDIX A

PROOF OF THEOREM 2

In this appendix, we prove theorem 2. Recall that $\Pr[\text{err}|\boldsymbol{x}]$ is the probability that the LP decoder fails, given that $\boldsymbol{x}$ was transmitted. It suffices to show that

$$\Pr[\text{err}|\boldsymbol{x}] = \Pr[\text{err}|\boldsymbol{0}], \tag{16}$$

for an arbitrary transmitted codeword $\boldsymbol{x} \in \mathcal{C}$. Define $B(\boldsymbol{x})$ as follows.

$$B(\boldsymbol{x}) = \left\{ \overset{\leftrightarrow}{\boldsymbol{y}} \in (\Sigma^2)^n \mid \exists (\boldsymbol{x}', \boldsymbol{\tau}') \text{ s.t. } (\boldsymbol{x}', \boldsymbol{\tau}') \text{ is a feasible solution of (4), } \boldsymbol{x} \neq \boldsymbol{x}', \langle \boldsymbol{\lambda}, \boldsymbol{\tau}' \rangle \leq \langle \boldsymbol{\lambda}, T(\boldsymbol{x}) \rangle \right\}. \tag{17}$$

Note that $\boldsymbol{\lambda}$ is a function of the received vector $\overset{\leftrightarrow}{\boldsymbol{y}}$. $B(\boldsymbol{x})$ is the set of received vectors $\overset{\leftrightarrow}{\boldsymbol{y}}$ that cause decoding failure when $\boldsymbol{x}$ was transmitted. We can rewrite (5) as follows.

$$\Pr[\text{err}|\boldsymbol{x}] = \sum_{\overset{\leftrightarrow}{\boldsymbol{y}} \in B(\boldsymbol{x})} p(\overset{\leftrightarrow}{\boldsymbol{y}}|\boldsymbol{x}). \tag{18}$$

Applying the above to the codeword $\boldsymbol{0}$, we obtain

$$\Pr[\text{err}|\boldsymbol{0}] = \sum_{\overset{\leftrightarrow}{\boldsymbol{y}} \in B(\boldsymbol{0})} p(\overset{\leftrightarrow}{\boldsymbol{y}}|\boldsymbol{0}). \tag{19}$$

In a similar manner to the proof in [6], we will show that the space $(\Sigma^2)^n$ of received vectors can be partitioned into pairs $(\overset{\leftrightarrow}{\boldsymbol{y}}, \overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}})$ such that

$$\Pr[\overset{\leftrightarrow}{\boldsymbol{y}}|\boldsymbol{x}] = \Pr[\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}}|\boldsymbol{0}], \tag{20}$$

$$\overset{\leftrightarrow}{\boldsymbol{y}} \in B(\boldsymbol{x}) \Leftrightarrow \overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}} \in B(\boldsymbol{0}). \tag{21}$$

These, along with (18) and (19), give

$$\Pr[\text{err}|\boldsymbol{x}] = \Pr[\text{err}|\boldsymbol{0}].$$

We partition the space of received vectors as follows. For an arbitrary received vector $\overset{\leftrightarrow}{\boldsymbol{y}}$, define $\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}}$ as follows. Let $(\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}})_i$ be $(\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}})_i = (\overset{\leftrightarrow}{\boldsymbol{y}})_i \oplus_p (\pi(\boldsymbol{x}))_i$, where $\oplus_p$ denotes the pairwise exclusive OR, that is, $(a_1, b_1) \oplus_p (a_2, b_2) = (a_1 \oplus b_1, a_2 \oplus b_2)$. Then $(\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}})_i = (0, 0)$ if and only if $(\overset{\leftrightarrow}{\boldsymbol{y}})_i = (\pi(\boldsymbol{x}))_i$ and therefore,

$$\Pr[\overset{\leftrightarrow}{\boldsymbol{y}}|\boldsymbol{x}] = \Pr[\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}}|\boldsymbol{0}], \tag{22}$$

holds. Note that the operation to make $\overset{\leftrightarrow}{\boldsymbol{y}}_0$ from $\overset{\leftrightarrow}{\boldsymbol{y}}$ is its own inverse and it gives a valid partition of $(\Sigma^2)^n$ into pairs.

Now it remains to show that $\overset{\leftrightarrow}{\boldsymbol{y}} \in B(\boldsymbol{x})$ if and only if $\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}} \in B(\boldsymbol{0})$. Before proving it, we introduce some lemmas.

Lemma 2: Let $(\boldsymbol{x}^f, \boldsymbol{\tau}^f)$ be a feasible solution of (4). We define the relative solution $(\boldsymbol{x}^r, \boldsymbol{\tau}^r)$ of $(\boldsymbol{x}^f, \boldsymbol{\tau}^f)$ to $\boldsymbol{x} \in \mathcal{C}$ as follows.

$$\tau^r_{i,(a,b)} = \tau^f_{i,(a,b) \oplus_p (\pi(\boldsymbol{x}))_i}, \quad i = 0, \ldots, n-1, (a,b) \in \Sigma^2, \tag{23}$$

$$x^r_i = \tau^r_{i,(1,0)} + \tau^r_{i,(1,1)}, \quad i = 0, \ldots, n-1. \tag{24}$$

Then the relative solution $(\boldsymbol{x}^r, \boldsymbol{\tau}^r)$ is a feasible solution of (4).

Proof: $\tau^r_{i,(a,b)} \in [0,1]$ for $i = 0, \ldots, n-1, (a,b) \in \Sigma^2$ and $\sum_{(a,b)\in\Sigma^2} \tau^r_{i,(a,b)} = 1$ for $i = 0, \ldots, n-1$ are obvious. From the definition of the relative solution,

$$\tau^r_{i,(0,1)} + \tau^r_{i,(1,1)} = \tau^f_{i,(0,1)\oplus_p(\pi(\boldsymbol{x}))_i} + \tau^f_{i,(1,1)\oplus_p(\pi(\boldsymbol{x}))_i}$$

$$= \begin{cases} \tau^f_{i,(0,1)} + \tau^f_{i,(1,1)} & \text{if } (\pi(\boldsymbol{x}))_i = (0,0) \text{ or } (1,0) \\ \tau^f_{i,(0,0)} + \tau^f_{i,(1,0)} = 1 - (\tau^f_{i,(0,1)} + \tau^f_{i,(1,1)}) & \text{if } (\pi(\boldsymbol{x}))_i = (0,1) \text{ or } (1,1), \end{cases} \tag{25}$$

$$x^r_{i+1} = \tau^r_{i+1,(1,0)} + \tau^r_{i+1,(1,1)} = \tau^f_{i+1,(1,0)\oplus_p(\pi(\boldsymbol{x}))_{i+1}} + \tau^f_{i+1,(1,1)\oplus_p(\pi(\boldsymbol{x}))_{i+1}}$$

$$= \begin{cases} \tau^f_{i+1,(1,0)} + \tau^f_{i,(1,1)} & \text{if } (\pi(\boldsymbol{x}))_{i+1} = (0,0) \text{ or } (0,1) \\ \tau^f_{i+1,(0,0)} + \tau^f_{i+1,(0,1)} = 1 - (\tau^f_{i+1,(1,0)} + \tau^f_{i+1,(1,1)}) & \text{if } (\pi(\boldsymbol{x}))_{i+1} = (1,0) \text{ or } (1,1). \end{cases}$$

$$\tag{26}$$

Considering the fact that the second element of $(\pi(\boldsymbol{x}))_i$ equals to the first element of $(\pi(\boldsymbol{x}))_{i+1}$, it always holds that $\tau^r_{i,(0,1)} + \tau^r_{i,(1,1)} = \tau^r_{i+1,(1,0)} + \tau^r_{i+1,(1,1)} = x^r_{i+1}$ from $x_{i+1} = \tau^f_{i,(0,1)} + \tau^f_{i,(1,1)} = \tau^f_{i+1,(1,0)} + \tau^f_{i+1,(1,1)}$. Finally, we show that $\boldsymbol{x}^r \in \mathcal{Q}(H)$. From the definition,

$$x^r_i = \tau^r_{i,(1,0)} + \tau^r_{i,(1,1)}$$

$$= \begin{cases} \tau^f_{i,(1,0)} + \tau^f_{i,(1,1)} = x^f_i & \text{if } (\pi(\boldsymbol{x}))_i = (0,0) \text{ or } (0,1) \\ \tau^f_{i,(0,0)} + \tau^f_{i,(0,1)} = 1 - x^f_i & \text{if } (\pi(\boldsymbol{x}))_i = (1,0) \text{ or } (1,1) \end{cases}$$

$$= |x^f_i - x_i|. \tag{27}$$

It is proved in [6] that $\boldsymbol{x}^r$, whose elements are calculated from (27), satisfies $\boldsymbol{x}^r \in \mathcal{Q}(H)$. ∎

For the relative solutions, following lemma holds.

Lemma 3: For an arbitrary $\boldsymbol{x} \in \mathcal{C}$, an arbitrary feasible solution $(\boldsymbol{x}^f, \boldsymbol{\tau}^f)$, and an arbitrary received vector $\overset{\leftrightarrow}{\boldsymbol{y}}$, it holds that

$$\langle \boldsymbol{\lambda}, \boldsymbol{\tau}^f - \boldsymbol{\tau} \rangle = \langle \boldsymbol{\lambda}^0, \boldsymbol{\tau}^r - \boldsymbol{\tau}^0 \rangle, \tag{28}$$

where $\boldsymbol{\lambda} = \Lambda(\overset{\leftrightarrow}{\boldsymbol{y}})$, $\boldsymbol{\tau} = T(\boldsymbol{x})$, $\boldsymbol{\lambda}^0 = \Lambda(\overset{\leftrightarrow}{\boldsymbol{y}_0})$, $\boldsymbol{\tau}^0 = T(\boldsymbol{0})$, and $\boldsymbol{\tau}^r$ is the relative solution of $\boldsymbol{\tau}^f$ to $\boldsymbol{x}$.

Proof: From the definition of $\boldsymbol{\lambda}^0$, it holds that

$$\lambda^0_{i,(a,b)} = -\ln p(\overset{\leftrightarrow^0}{\boldsymbol{y}}_i | (a,b))$$

$$= -\ln p(\overset{\leftrightarrow}{\boldsymbol{y}}_i \oplus_p (\pi(\boldsymbol{x}))_i | (a,b))$$

$$= -\ln p(\overset{\leftrightarrow}{\boldsymbol{y}}_i | (a,b) \oplus_p (\pi(\boldsymbol{x}))_i)$$

$$= \lambda_{i,(a,b)\oplus_p(\pi(\boldsymbol{x}))_i}. \tag{29}$$

Therefore, for $i = 0, \ldots, n-1$, it holds that

$$\sum_{(a,b)\in\Sigma^2} \lambda^0_{i,(a,b)} \tau^r_{i,(a,b)} = \sum_{(a,b)\in\Sigma^2} \lambda_{i,(a,b)\oplus_p(\pi(\boldsymbol{x}))_i} \tau^f_{i,(a,b)\oplus_p(\pi(\boldsymbol{x}))_i} \tag{30}$$

$$= \sum_{(a,b)\in\Sigma^2} \lambda_{i,(a,b)} \tau^f_{i,(a,b)}, \tag{31}$$

and $\langle \boldsymbol{\lambda}, \boldsymbol{\tau}^f \rangle = \langle \boldsymbol{\lambda}^0, \boldsymbol{\tau}^r \rangle$. Further, from $\sum_{(a,b) \in \Sigma^2} \lambda_{i,(a,b)} \tau_{i,(a,b)} = \lambda_{i,(\pi(\boldsymbol{x}))_i}$, $\sum_{(a,b) \in \Sigma^2} \lambda^0_{i,(a,b)} \tau^0_{i,(a,b)} = \lambda^0_{i,(0,0)}$ and $\lambda^0_{i,(0,0)} = \lambda_{i,(\pi(\boldsymbol{x}))_i}$, it also holds that $\langle \boldsymbol{\lambda}, \boldsymbol{\tau} \rangle = \langle \boldsymbol{\lambda}^0, \boldsymbol{\tau}^0 \rangle$. ∎

We can prove the following lemma from Lemma 2 and Lemma 3.

Lemma 4: Fix a codeword $\boldsymbol{x} \in \mathcal{C}$. For an arbitrary feasible solution $(\boldsymbol{x}^f, \boldsymbol{\tau}^f)$ of (4), $\boldsymbol{x}^f \neq \boldsymbol{x}$, there exists another feasible solution $(\boldsymbol{x}^r, \boldsymbol{\tau}^r)$ such that $\boldsymbol{x}^r \neq \boldsymbol{0}$ and

$$\langle \boldsymbol{\lambda}, \boldsymbol{\tau}^f - \boldsymbol{\tau} \rangle = \langle \boldsymbol{\lambda}^0, \boldsymbol{\tau}^r - \boldsymbol{\tau}^0 \rangle. \tag{32}$$

Further, for an arbitrary feasible solution $(\boldsymbol{x}^r, \boldsymbol{\tau}^r)$ of (4), $\boldsymbol{x}^r \neq \boldsymbol{0}$, there exists another feasible solution $(\boldsymbol{x}^f, \boldsymbol{\tau}^f)$ such that $\boldsymbol{x}^f \neq \boldsymbol{x}$ and

$$\langle \boldsymbol{\lambda}^0, \boldsymbol{\tau}^r - \boldsymbol{\tau}^0 \rangle = \langle \boldsymbol{\lambda}, \boldsymbol{\tau}^f - \boldsymbol{\tau} \rangle. \tag{33}$$

Proof: Let $(\boldsymbol{x}^r, \boldsymbol{\tau}^r)$ be the relative solution of $(\boldsymbol{x}^f, \boldsymbol{\tau}^f)$ to $\boldsymbol{x}$. From Lemma 2, $(\boldsymbol{x}^r, \boldsymbol{\tau}^r)$ is a feasible solution of (4). From the definition, $\boldsymbol{x}^r \neq \boldsymbol{0}$ and applying Lemma 3, we obtain

$$\langle \boldsymbol{\lambda}, \boldsymbol{\tau}^f - \boldsymbol{\tau} \rangle = \langle \boldsymbol{\lambda}^0, \boldsymbol{\tau}^r - \boldsymbol{\tau}^0 \rangle. \tag{34}$$

The latter of the lemma can be proved by replacing the role of $(\boldsymbol{x}^f, \boldsymbol{\tau}^f)$ and $(\boldsymbol{x}^r, \boldsymbol{\tau}^r)$. ∎

We are now ready to prove that $\overset{\leftrightarrow}{\boldsymbol{y}} \in B(\boldsymbol{x})$ if and only if $\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}} \in B(\boldsymbol{0})$ and it completes the proof of the Theorem 2. Assume that $\overset{\leftrightarrow}{\boldsymbol{y}} \in B(\boldsymbol{x})$. From the definition of $B(\boldsymbol{x})$, there exists a feasible solution $(\boldsymbol{x}^f, \boldsymbol{\tau}^f)$ of (4) that satisfies $\boldsymbol{x}^f \neq \boldsymbol{x}$ and

$$\langle \boldsymbol{\lambda}, \boldsymbol{\tau}^f - \boldsymbol{\tau} \rangle \leq 0. \tag{35}$$

From Lemma 4, there exists a feasible solution $(\boldsymbol{x}^r, \boldsymbol{\tau}^r)$ such that $\boldsymbol{x}^r \neq \boldsymbol{0}$ and

$$\langle \boldsymbol{\lambda}^0, \boldsymbol{\tau}^r - \boldsymbol{\tau}^0 \rangle \leq 0. \tag{36}$$

It means $\overset{\leftrightarrow}{\boldsymbol{y}}_{\boldsymbol{0}} \in B(\boldsymbol{0})$. The other direction can be proved with the symmetric argument.

APPENDIX B

PROOF OF LEMMA 1

It is known that a vertex of a polytope of dimension N is uniquely determined by N linearly independent facets of the polytope which the vertex sits on [12]. The dimension of the feasible region of (4) is $5n$. Let $(\boldsymbol{x}, \boldsymbol{\tau})$ be an arbitrary vertex of the feasible region of (4) and hence it sits on $5n$ linearly independent facets in (4). We will show that $(\boldsymbol{x}, \boldsymbol{\tau})$ sits on at most $4n$ linearly independent facets in (4) which are not included in $\boldsymbol{x} \in \mathcal{Q}(H)$. It leads that $\boldsymbol{x}$ sits on at least n linearly independent facets in $\boldsymbol{x} \in \mathcal{Q}(H)$. Since $\boldsymbol{x}$ satisfies at most n linearly independent equality, $\boldsymbol{x}$ sits on just n linearly independent facets in $\boldsymbol{x} \in \mathcal{Q}(H)$ and therefore $\boldsymbol{x} \in \mathcal{V}_{\mathcal{Q}}$.

Here we show that an arbitrary vertex $(\boldsymbol{x}, \boldsymbol{\tau})$ of the feasible region of (4) sits on at most $4n$ linearly independent facets in (4) other than the constraints in $\boldsymbol{x} \in \mathcal{Q}(H)$. Let E_1, E_2, E_3, E_4 be the sets which are defined as

$$E_1 = \{i \in \{1, \ldots, n\} \mid x_i \text{ and } x_{i+1} \text{ are integral}\}, \tag{37}$$

$$E_2 = \{i \in \{1, \ldots, n\} \mid x_i \text{ is integral}, x_{i+1} \text{ is not integral}\}, \tag{38}$$

$$E_3 = \{i \in \{1, \ldots, n\} \mid x_i \text{ is not integral}, x_{i+1} \text{ is integral}\}, \tag{39}$$

$$E_4 = \{i \in \{1, \ldots, n\} \mid x_i \text{ and } x_{i+1} \text{ are not integral}\}. \tag{40}$$

We show that for any $i \in E_j, j = 1, \ldots, 4$, $\{\tau_{i,(a,b)}\}_{(a,b)\in\Sigma^2}$, x_i, and x_{i+1} satisfy at most 4 linearly independent constraints in (4) other than the constraints in $\boldsymbol{x} \in \mathcal{Q}(H)$.

In the case $i \in E_1$, $\tau_{i,(x_i,x_{i+1})} = 1$ and $\tau_{i,(a,b)} = 0, (a,b) \neq (x_i, x_{i+1})$. Other constraints are not linearly independent from these constraints and $x_i, x_{i+1} \in \{0, 1\}$. Therefore just 4 linearly independent equality constraints $\tau_{i,(a,b)} \in \{0, 1\}, (a, b) \in \Sigma^2$ are satisfied.

Next, we consider the case $i \in E_2$. First we assume that $x_i = 0$. Then $\tau_{i,(1,0)} = \tau_{i,(1,1)} = 0$ holds from the constraint $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$. In this case, it holds that $0 < \tau_{i,(0,0)}, \tau_{i,(0,1)} < 1$ otherwise it leads contradiction to the assumption x_{i+1} is not integral (For example, if $\tau_{i,(0,0)} = 0$ then $\tau_{i,(0,1)} = 1$ and $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)} = 1$). The equality $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ is linearly dependent to the equalities $\tau_{i,(1,0)} = 0$, $\tau_{i,(1,1)} = 0$ and $x_i = 0$. Therefore just 4 linearly independent equality constraints $\tau_{i,(1,0)} = 0$, $\tau_{i,(1,1)} = 0$, $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$, $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}$ are satisfied. When $x_i = 1$ then $\tau_{i,(0,0)} = \tau_{i,(0,1)} = 0$ holds from the constraints $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ and $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$. It can be shown that just 4 linearly independent equality constraints are satisfied in a similar way for the case $x_i = 0$.

For the case $i \in E_3$, we can prove that just 4 linearly independent equality constraints are satisfied in a similar way for the case $i \in E_2$ by exchanging the role of x_i and x_{i+1}.

Finally, we consider the case $i \in E_4$. From the assumption that $0 < x_i < 1$ and constraint $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$, it turned out that one of the three cases a) $\tau_{i,(1,0)} = 0, 0 < \tau_{i,(1,1)} < 1$ or b) $\tau_{i,(1,1)} = 0, 0 < \tau_{i,(1,0)} < 1$ or c) $0 < \tau_{i,(1,0)}, \tau_{i,(1,1)} < 1$ occurs.

a)   We consider the case that $\tau_{i,(1,0)} = 0, 0 < \tau_{i,(1,1)} < 1$ hold. It holds that $\tau_{i,(0,0)} + \tau_{i,(0,1)} = 1 - x_i$ from the constraints $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ and $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$. One of the three cases a-1) $\tau_{i,(0,0)} = 0$ or a-2) $\tau_{i,(0,1)} = 0$ or a-3) $0 < \tau_{i,(0,0)}, \tau_{i,(0,1)} < 1$ occurs.

    a-1)   If we assume $\tau_{i,(0,0)} = 0$ then $\tau_{i,(0,1)} = 1 - x_i$ and $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)} = 1$. But it contradicts the assumption x_i is not integral and it never happens.

    a-2)   We assume $\tau_{i,(0,1)} = 0$. From the constraint $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ and the assumption $\tau_{i,(1,0)} = 0$, $\tau_{i,(1,1)} = x_i$. Therefore, $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)} = 0 + x_i = x_i$ holds. Above argument shows that the constraint $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}$ is linearly dependent to the constraints $\tau_{i,(1,0)} = 0$, $\tau_{i,(0,1)} = 0$, $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ and just 4 linearly independent constraints $\tau_{i,(1,0)} = 0$, $\tau_{i,(0,1)} = 0$, $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$, $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ are satisfied.

a-3) When $0 < \tau_{i,(0,0)}, \tau_{i,(0,1)} < 1$ holds, 4 linearly independent constraints $\tau_{i,(1,0)} = 0$, $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$, $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$, $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}$ are satisfied.

b) We consider the case that $\tau_{i,(1,1)} = 0$, $0 < \tau_{i,(1,0)} < 1$ hold. As in the case a), it holds that $\tau_{i,(0,0)} + \tau_{i,(0,1)} = 1 - x_i$ and one of the three cases b-1) $\tau_{i,(0,0)} = 0$ or b-2) $\tau_{i,(0,1)} = 0$ or b-3) $0 < \tau_{i,(0,0)}, \tau_{i,(0,1)} < 1$ occurs.

b-1) Consider the case that $\tau_{i,(0,0)} = 0$ holds. Then $\tau_{i,(0,1)} = 1 - x_i$ and $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)} = 1 - x_i$ holds. Above argument shows that the constraint $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}$ is linearly dependent to the constraints $\tau_{i,(0,0)} = 0$, $\tau_{i,(1,1)} = 0$ and $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$. Therefore just 4 linearly independent constraints $\tau_{i,(0,0)} = 0$, $\tau_{i,(1,1)} = 0$, $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$ and $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ are satisfied.

b-2) If we assume that $\tau_{i,(0,1)} = 0$ holds then $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)} = 0$, but it contradict the assumption x_{i+1} is not integral. Therefore this case never occurs.

b-3) When $0 < \tau_{i,(0,0)}, \tau_{i,(0,1)} < 1$ holds, 4 linearly independent constraints $\tau_{i,(1,1)} = 0$, $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$, $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$, $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}$ are satisfied.

c) Assume that $0 < \tau_{i,(1,0)}, \tau_{i,(1,1)} < 1$ holds. As in the other cases, it holds that $\tau_{i,(0,0)} + \tau_{i,(0,1)} = 1 - x_i$ and one of the three cases c-1) $\tau_{i,(0,0)} = 0$ or c-2) $\tau_{i,(0,1)} = 0$ or c-3) $0 < \tau_{i,(0,0)}, \tau_{i,(0,1)} < 1$ occurs.

c-1) When $\tau_{i,(0,0)} = 0$ holds then $\tau_{i,(0,1)} = 1 - x_i$ and therefore $0 < \tau_{i,(0,1)} < 1$. In this case, 4 linearly independent constraints $\tau_{i,(0,0)} = 0$, $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$, $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ and $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}$ are satisfied.

c-2) When $\tau_{i,(0,1)} = 0$ holds then $\tau_{i,(0,0)} = 1 - x_i$ and $0 < \tau_{i,(0,0)} < 1$ hold. Therefore 4 linearly independent constraints $\tau_{i,(0,1)} = 0$, $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$, $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ and $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}$ are satisfied.

c-3) Assume that $0 < \tau_{i,(0,0)}, \tau_{i,(0,1)} < 1$ holds. In this case, 3 linearly independent constraints $\sum_{(a,b)\in\Sigma^2} \tau_{i,(a,b)} = 1$, $x_i = \tau_{i,(1,0)} + \tau_{i,(1,1)}$ and $x_{i+1} = \tau_{i,(0,1)} + \tau_{i,(1,1)}$ are satisfied.

REFERENCES

[1] Y. Cassuto and M. Blaum, "Codes for symbol-pair read channels," IEEE Trans. on Inf. Theory, vol. 57, no. 12, pp. 8011-8020, Dec. 2011.

[2] Y. Cassuto and S. Litsyn, "Symbol-pair codes: algebraic constructions and asymptotic bounds," Proc. of 2011 IEEE International Symposium on Information Theory, pp. 2348-2352, 2011.

[3] E. Yaakobi, J. Bruck, P. H. Siegel, "Decoding of Cyclic Codes over Symbol-Pair Read Channels," Proc. of 2012 IEEE Int. Symp. on Inf. Theory, pp. 2891-2895, 2012.

[4] Y. M. Chee, H. Kiah, and C. Wang, "Maximum distance separable symbol-pair codes," Proc. of 2012 IEEE Int. Symp. on Inf. Theory, pp. 2896-2900, 2012.

[5] M. Hirotomo, M. Takita, and M. Morii, "Syndrome decoding of symbol-pair codes," Proc. of 2014 IEEE Inf. Theory Workshop, pp.163-167, 2014.

[6] J. Feldman, M. J. Wainwright, D. R. Karger, "Using Linear Programming to Decode Binary Linear Codes," IEEE Trans. on Inf. Theory, vol. 51, no. 3, pp. 954-972, 2005.

[7] J. Feldman and C. Stein, "LP decoding achieves capacity," Proc. of the 16th annual ACM-SIAM symposium on Discrete Algorithms, pp. 460-469, 2005.

[8] M. H. Taghavi and P. H. Siegel, "Adaptive methods for linear programming decoding," IEEE Trans. Inf. Theory, vol. 54, no. 12, pp. 5396-5410, 2008.

[9] X. Zhang and P. H. Siegel, "Efficient Iterative LP Decoding of LDPC Codes with Alternating Direction Method of Multipliers," Proc. of 2013 IEEE Int. Symp. on Inf. Theory, pp. 1501-1505, 2013.

[10] R. Gallager, "Low-density parity-check codes," IRE Trans. Inf. Theory, vol. IT-8, no. 1, pp. 21-28, Jan. 1962.

[11] E. Berlekamp, R. J. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," IEEE Trans. Inf. Theory, vol. IT-24, no. 3, pp. 384-386, May 1978.

[12] A. Schrijver, "Theory of Linear and Integer Programming," New York: Wiley, 1987.

[13] B. H. Kim, H. D. Pfister, "Joint Decoding of LDPC Codes and Finite-State Channels via Linear Programming," IEEE Journal of Selected Topics in Signal Processing, Vol. 5, No. 8, pp. 1563-1576, 2011.

[14] M. F. Flanagan, V. Skachek, E. Byrne, M. Greferath, "Linear-Programming Decoding of Nonbinary Linear Codes," IEEE Trans. on Inf. Theory, Vol. 55, No. 9, pp.4134-4154, 2009.